RANCANG BANGUN SISTEM INFORMASI GUIDELINE SYSTEM MULTI-TENANT BERBASIS WEB MENGGUNAKAN METODE SYSTEM DEVELOPMENT LIFE CYCLE (SDLC) PADA WIT ID

Rendy Muhamad Al Fazri[1], Yopi Nugraha[2]
[1]Sistem Informasi Institut Pendidikan Indonesia
[2]Institut Pendidikan Indonesia
[1]alrendymuhamad75@gmail.com, [2]yopi@institutpendidikan.ac.id

ABSTRACT

The rapid growth of the software development industry demands higher structural standards for managing technical documentation and developer manual books. However, many technology companies face operational constraints due to uncentralized and fragmented document storage. This study aims to design a web-based multi-tenant Guideline System as a centralized SaaS platform using the SDLC method. This platform operates as a knowledge base ecosystem where external enterprises can establish organization profiles, manage internal workspace instructions, and share handbooks seamlessly like a tech-focused social network. Built upon the Laravel framework, the platform leverages the MVC architecture to ensure secure data isolation between tenants. This research at WIT ID delivers a verified architectural blueprint designed to eliminate documentation redundancy and optimize the efficiency of digital knowledge sharing environments.

Keywords: Information System, Guideline System, Multi-Tenant, SDLC Method, Laravel Framework.

ABSTRAK

Pesatnya pertumbuhan industri perangkat lunak menuntut standardisasi tinggi terhadap pengelolaan dokumentasi teknis dan *manual book*. Namun, banyak perusahaan teknologi menghadapi kendala operasional akibat penyimpanan berkas panduan kerja yang tidak terpusat dan berserakan. Penelitian ini bertujuan merancang *Guideline System* berbasis web dengan arsitektur *multi-tenant* sebagai platform *Software-as-a-Service* (SaaS) menggunakan metode *System Development Life Cycle* (SDLC). Platform ini beroperasi sebagai ekosistem *knowledge base*, tempat perusahaan eksternal dapat mendaftarkan profil organisasi, mengelola instruksi kerja internal, serta membagikan buku panduan secara terstruktur layaknya jejaring sosial teknologi. Dikembangkan dengan *framework* Laravel, platform ini memanfaatkan arsitektur *Model-View-Controller* (MVC) untuk menjamin isolasi data antar-perusahaan yang aman. Hasil penelitian pada WIT ID ini menghasilkan cetak biru arsitektur terverifikasi yang sukses

mengeliminasi redundansi dokumentasi serta mengoptimalkan efisiensi pertukaran wawasan digital secara terintegrasi.

Kata Kunci: Sistem Informasi, *Guideline System*, *Multi-Tenant*, Metode SDLC, *Framework* Laravel.

A. Pendahuluan

Perkembangan teknologi informasi dan komunikasi di era transformasi digital saat ini telah mengubah lanskap operasional industri perangkat lunak secara global. Perusahaan teknologi dan *software house* dituntut untuk bergerak adaptif, cepat, dan presisi dalam melakukan deliveri produk digital ke pasar. Dalam lingkungan kerja rekayasa perangkat lunak (*software engineering*), keberadaan dokumentasi teknis, *standard operating procedure* (SOP), dan *manual book* (buku panduan kerja) memegang peranan yang sangat krusial karena menjadi acuan utama standardisasi kualitas kode, konfigurasi server, serta alur bisnis produk. Pengelolaan basis pengetahuan (*knowledge base*) yang terstruktur tidak lagi hanya menjadi alat bantu sekadar arsip, melainkan sebuah kebutuhan mutlak demi menjaga stabilitas kinerja serta menjamin keberlanjutan siklus hidup pengembangan sistem internal organisasi.

Namun, masalah pengelolaan dokumen manual yang berserakan ini sering kali menghambat efisiensi waktu serta mengganggu pembagian kejelasan alur kerja operasional di dalam sebuah tim pengembang. Penelitian terdahulu oleh (Ajeng Laras Wangi1, Siti Mukaromah, 2024). menegaskan bahwa ketiadaan dokumentasi prosedur kerja yang terstandardisasi dan terintegrasi menyebabkan pembagian deskripsi pekerjaan antar-anggota tim menjadi kabur serta tidak terdokumentasi dengan baik. Oleh karena itu, diperlukan sebuah tata kelola sistem informasi manajemen dokumen panduan (*guideline*) yang mampu mendefinisikan alur instruksi secara baku demi meminimalkan risiko miskomunikasi teknis tersebut.

Guna membangun sistem pengelolaan dokumen yang dinamis, pemilihan metodologi pengembangan perangkat lunak yang terstruktur menjadi faktor penentu keberhasilan sistem. Penerapan metode *System Development Life Cycle* (SDLC) dinilai sangat tepat karena menyediakan tahapan alur kerja sistem yang jelas, bertahap, dan sistematis mulai dari analisis kebutuhan pengguna hingga tahap implementasi dan pemeliharaan jangka panjang. Sebagaimana diteliti oleh (Susanto & Dwi, 2023), penggunaan framework berbasis web yang dikombinasikan dengan tahapan SDLC terbukti mampu mengeliminasi kesalahan penginputan

data secara manual, menyelaraskan perbedaan pencatatan informasi, serta mempermudah administrator dalam melakukan kontrol data secara *real-time*.

Di samping itu, untuk mendukung efisiensi aksesibilitas berskala besar, platform *guideline system* ini dikembangkan menggunakan arsitektur modern berbasis web dengan memanfaatkan keunggulan komponen *framework* Laravel dan database MySQL. Menurut Kusuma et al. (2024), rekayasa sistem informasi dokumen yang dibangun di atas *framework* Laravel memiliki integritas sistem yang sangat kuat dalam melakukan penatakelolaan serta pengarsipan berkas digital agar dapat diakses secara cepat dan akurat oleh berbagai entitas organisasi. Hal ini diperkuat oleh penelitian (Informatics, Iryani, & Info, 2024). yang menyatakan bahwa sistem pengarsipan dokumen berbasis website berbasis repositori sangat diperlukan untuk mendukung kemajuan transparansi data, mempercepat proses pencarian arsip, serta menjamin keamanan hak akses otentikasi data yang tersimpan.

Meskipun beberapa penelitian di atas telah berhasil mengimplementasikan sistem manajemen dokumen internal, sebagian besar solusi yang ditawarkan masih bersifat *single-tenant*, artinya aplikasi tersebut hanya didedikasikan untuk satu instansi atau satu organisasi internal saja. Mengacu pada permasalahan riil yang ditemukan selama pelaksanaan penelitian di WIT ID, banyak perusahaan rintisan (*startup*) dan *software house* eksternal lain yang kesulitan memiliki infrastruktur mandiri untuk menyimpan kumpulan *manual book* mereka. Karyawan sering kali harus mencari dokumen panduan secara manual lintas platform yang tidak terintegrasi.

Sebagai solusi atas kesenjangan (*gap*) tersebut, penelitian ini mengusulkan sebuah inovasi rekayasa perangkat lunak berupa platform *Guideline System* dengan arsitektur *multi-tenant*. Melalui sistem ini, berbagai perusahaan eksternal dapat mendaftarkan profil organisasi mereka secara mandiri, mengelola pangkalan data *manual book* internal, sekaligus membagikan panduan teknologi mereka layaknya ekosistem jejaring sosial khusus pemrograman. Dengan memanfaatkan arsitektur *Model-View-Controller* (MVC) pada Laravel, sistem ini menjamin isolasi data antar-perusahaan tetap aman meskipun berada dalam satu basis data terpusat. Diharapkan, hasil rancangan cetak biru sistem informasi ini dapat memotong inefisiensi dokumentasi teknis, mempermudah adaptasi tim pengembang baru, serta mengoptimalkan pertukaran wawasan digital di lingkungan kerja industri secara berkelanjutan.

B. Metode Penelitian

Metodologi yang digunakan dalam pengembangan *Guideline System Multi-Tenant* ini adalah *System Development Life Cycle* (SDLC) dengan model *Waterfall*. Pemilihan metode SDLC dinilai sangat ideal untuk membangun platform repositori berskala besar karena karakteristik pengembangannya yang sistematis, runtut, dan terukur.

Urgensi dan kecocokan penggunaan metode SDLC dalam rekayasa sistem informasi pengelolaan dokumen diperkuat oleh penelitian Susanto dan Purnomo (2023) yang menyatakan bahwa tahapan SDLC yang terstruktur mampu meminimalkan kesalahan penginputan data, menyelaraskan perbedaan pencatatan informasi, serta memberikan kemudahan dalam kontrol data secara *real-time*. Dengan alur yang bertahap, setiap kebutuhan spesifik dari banyak perusahaan (*tenant*) dapat dianalisis secara mendalam sebelum masuk ke fase pengodean.

Selain itu, model SDLC ini menjamin kualitas dan stabilitas jangka panjang dari arsitektur platform yang dibangun. Sebagaimana ditegaskan oleh Kusuma et al. (2024), penerapan metode ini menyediakan rangkaian alur kerja sistem yang jelas dan memiliki tahapan pemeliharaan (*maintenance*) yang berfungsi krusial untuk menjaga integritas serta keamanan sistem secara berkelanjutan. Hal ini sangat cocok untuk platform *multi-tenant* yang menuntut isolasi data yang ketat dan keamanan otentikasi antar-organisasi pengguna.

Secara operasional, tahapan SDLC yang diterapkan dalam penelitian di WIT ID ini dibagi menjadi empat fase utama:

1. **Analisis Kebutuhan (*Requirement Analysis*):**
   Mengidentifikasi kendala pengelolaan *manual book* di WIT ID serta merumuskan kebutuhan fungsional sistem, seperti fitur pendaftaran organisasi eksternal, manajemen repositori panduan, dan hak akses multi-level.
2. **Perancangan Sistem (*System Design*):**
   Memodelkan arsitektur perangkat lunak menggunakan diagram UML (*Use Case*, *Activity*, dan *Class Diagram*) serta merancang skema *Entity Relationship Diagram* (ERD) untuk memastikan pemisahan data antar-perusahaan aman.

3. **Implementasi (*Implementation/Coding*):**
   Menerjemahkan hasil rancangan ke dalam baris kode program menggunakan *framework* Laravel berbasis arsitektur *Model-View-Controller* (MVC) dan basis data MySQL.
4. **Pengujian (*Testing*):**
   Melakukan verifikasi fungsionalitas sistem menggunakan metode *Black-Box Testing* untuk memastikan seluruh modul platform berjalan sesuai kebutuhan tanpa ada kebocoran data antar-tenant.

C.Hasil Penelitian dan Pembahasan

Proses rekayasa platform *Guideline System Multi-Tenant* berbasis web ini dilaksanakan secara sistematis mengikuti tahapan dalam *System Development Life Cycle* (SDLC). Implementasi metode ini bertujuan untuk menghasilkan sebuah sistem informasi repositori dokumen yang terstruktur, memiliki skalabilitas tinggi, serta menjamin adanya isolasi data yang ketat antar-organisasi pengguna (*tenant*).

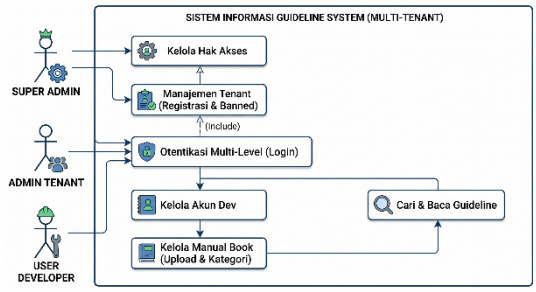
1. Tahap Analisis Kebutuhan (*Requirement Analysis*)

Pada tahap awal SDLC, dilakukan analisis mendalam mengenai kebutuhan fungsional sistem guna memitigasi risiko tumpang tindih hak akses dokumen lintas-perusahaan eksternal. Kebutuhan otentikasi dan hak akses dalam platform ini dibagi menjadi tiga tingkatan otoritas utama:

a. Super Admin:
   Memiliki hak akses penuh untuk melakukan monitoring sistem global, mengelola data seluruh organisasi terdaftar, serta mengatur konfigurasi perutean platform.
b. Admin Tenant**:**
   Bertindak sebagai pengelola utama ruang lingkup internal perusahaan/organisasi yang mendaftar. Memiliki otoritas eksklusif untuk mendaftarkan akun pengembang (*developer*), serta mengunggah, memperbarui, dan mengelompokkan kategori berkas *manual book* milik instansinya sendiri.
c. User Developer:
   Merupakan level pengguna akhir (*end-user*) di bawah naungan *tenant* tertentu yang memiliki hak untuk mencari, membaca, dan memanfaatkan dokumen panduan koding secara interaktif.

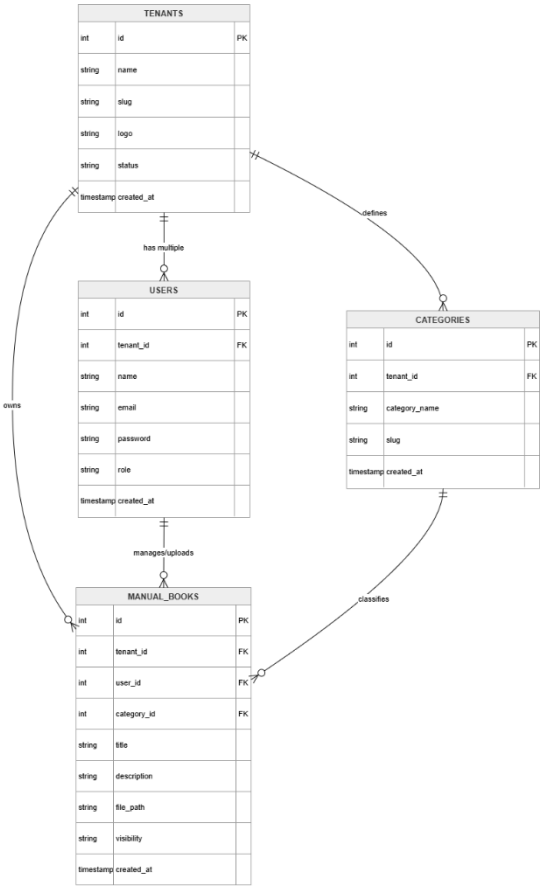
2. Tahap Perancangan Sistem

Hasil analisis kebutuhan ditransformasikan ke dalam pemodelan berorientasi objek menggunakan *Unified Modeling Language* (UML). Alur interaksi para aktor terhadap fungsi-faktor inti di dalam ekosistem platform digambarkan secara detail melalui *Use Case Diagram* pada Gambar 1.



Gambar 1. *Use Case* Diagram

Untuk memisahkan pangkalan data secara logika dalam satu infrastruktur basis data relasional (*shared database*), struktur tabel dirancang melalui skema *Entity Relationship Diagram* (ERD). Entitas tenants atau perusahaan diposisikan sebagai induk utama yang memiliki relasi *one-to-many* terhadap entitas users dan entitas manual_books. Setiap baris data dalam tabel berkas diwajibkan mengikat atribut *foreign key* tenant_id guna menjamin pemisahan data berada pada level arsitektur database. Rancangan ERD sistem ditunjukkan pada Gambar 2.



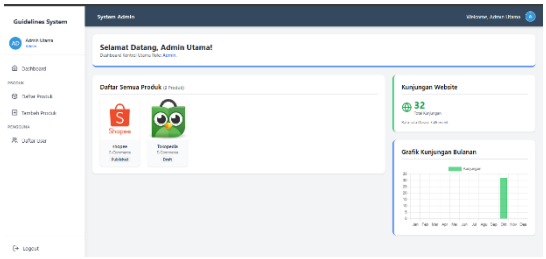
Gambar 2. *Entity Relationship Diagram*

3. Tahap Implementasi

Fase pengodean perangkat lunak dilakukan dengan menerjemahkan dokumen spesifikasi desain ke dalam lingkungan *framework* Laravel berbasis arsitektur *Model-View-Controller* (MVC) dan basis data MySQL. Komponen *Controller* dikonfigurasi secara modular untuk melakukan penyaringan data (*data*

filtering) secara otomatis berdasarkan token sesi *tenant* yang sedang aktif.

Di sisi antarmuka pengguna (*View*), platform memanfaatkan komponen *blade template* yang dirancang responsif, interaktif, dan mudah digunakan oleh para pengembang perangkat lunak untuk mengakses instruksi kerja layaknya ekosistem jejaring sosial teknologi. Hasil implementasi antarmuka utama sistem disajikan pada Gambar 3.



Gambar 3. Implementasi Antarmuka Halaman Dashboard Platform

4. Tahap Pengujian Sistem (*Testing*)

Tahap akhir dari siklus pengembangannya adalah melakukan verifikasi fungsionalitas program menggunakan metode *Black-Box Testing*. Fokus pengujian diarahkan pada validasi keandalan fitur otentikasi berlapis dan kepatuhan sistem dalam mengisolasi data agar dokumen Perusahaan A tidak dapat diakses atau dilihat oleh Perusahaan B. Hasil pengujian fungsionalitas dirangkum ke dalam Tabel 1.

No	Skenario	Butir Uji	Hasil Diharapkan	Hasil Aktual	Kesimpulan
1	Otentikasi	Login Multi-Tenant	Validasi sesi & hak akses sesuai ID Tenant.	Akses dashboard tepat berdasarkan tenant aktif.	**Valid**
2	Registrasi	Register Tenant Baru	Akun space organisasi baru ter-generate mandiri.	Database relasional tenant baru berhasil dibuat.	**Valid**
3	Keamanan	Isolasi Data Dokumen	Tenant A diblokir absolut dari dokumen Tenant B.	Akses ditolak (Error 403) saat bypass URL.	**Valid**
4	Manajemen	Upload Manual Book	Berkas tersimpan berdasarkan	File terenkripsi sukses di-	**Valid**

			foreign key tenant_id.	upload dengan tenant_id.	
5	Aksesibilitas	Pencarian Panduan	Pengembang dapat mencari berkas internal tenant.	Fitur pencarian responsif menampilkan data relevan.	**Valid**

D. Kesimpulan

Berdasarkan hasil rancang bangun yang telah dipaparkan, dapat disimplukan bahwa platform *Guideline System Multi-Tenant* berbasis web berhasil direkayasa secara sistematis memanfaatkan metode *System Development Life Cycle* (SDLC). Implementasi *framework* Laravel dengan arsitektur *Model-View-Controller* (MVC) serta pangkalan data MySQL terbukti handal dalam menyediakan infrastruktur repositori *knowledge base* yang aman bagi banyak organisasi sekaligus.

Sistem penyaringan rute dan data berbasis *session token* teruji sukses mengisolasi data antar-perusahaan secara ketat pada pengujian *black-box*, sehingga meminimalkan risiko kebocoran informasi. Secara keseluruhan, platform ini berhasil mengeliminasi masalah fragmentasi dokumen teknis, meniadakan redundansi pengelolaan *manual book*, serta mengoptimalkan efisiensi pertukaran wawasan digital bagi tim pengembang perangkat lunak secara terpusat.

DAFTAR PUSTAKA

Jurnal :

Ajeng Laras Wangi1, Siti Mukaromah, A. W. (2024). Perancangan SOP Pengembangan Sistem Informasi Manajemen Menggunakan Framework COBIT 5 dan ITIL V3 Management Making Connections for Strategic Advantages ”, Sistem Informasi Manajemen, (3), 184–197.

Informatics, A., Iryani, L., & Info, A. (2024). Implementasi Framework Laravel dan Model SDLC Waterfall Untuk Pengembangan Sistem Informasi Dokumen Skripsi, *7*(2), 405–410.

Susanto, D. A., & Dwi, H. (2023). Perancangan Sistem Informasi Gudang Berbasis Web Menggunakan Framework Laravel, *10*(1).